

Diagrams

Figure 1

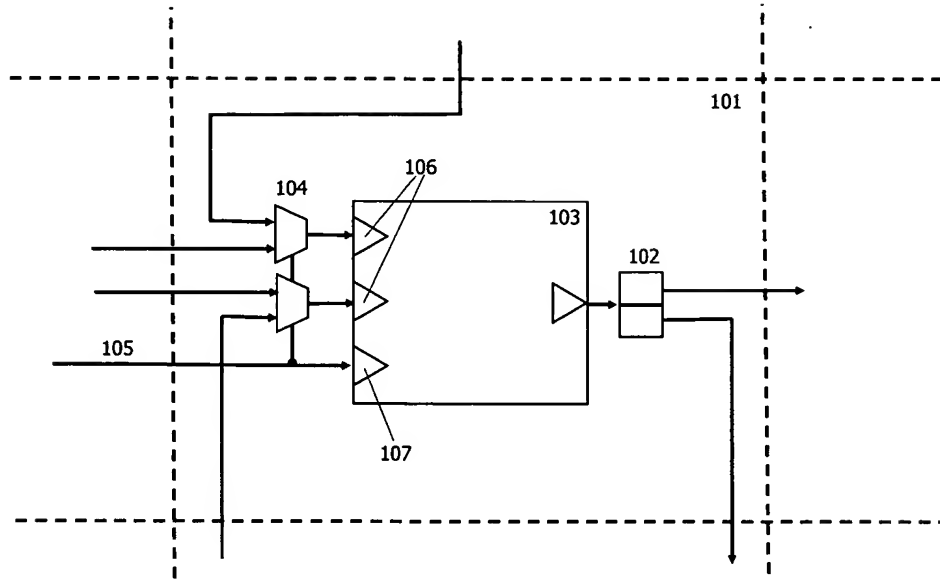


Figure 2

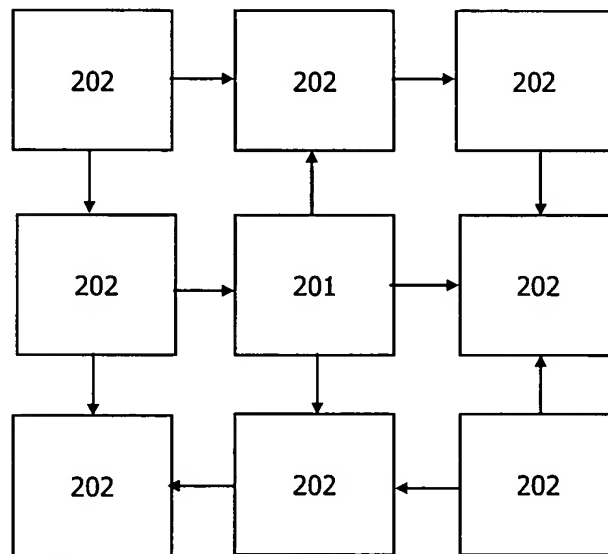


Figure 3

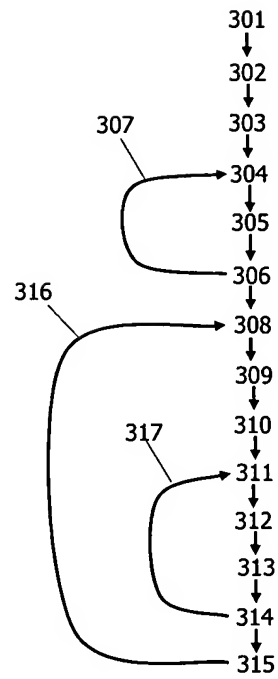


Figure 4

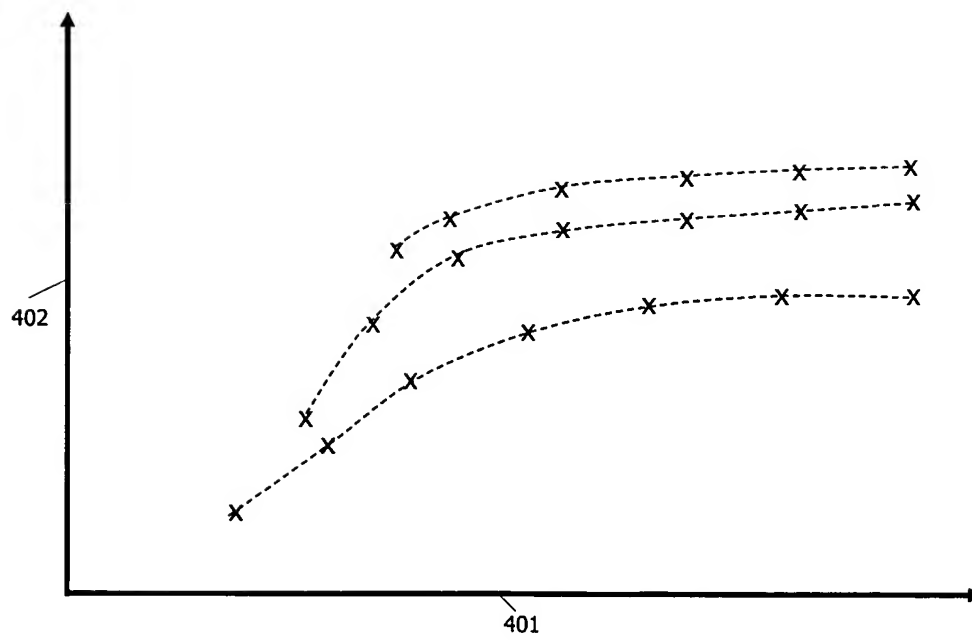


Figure 5

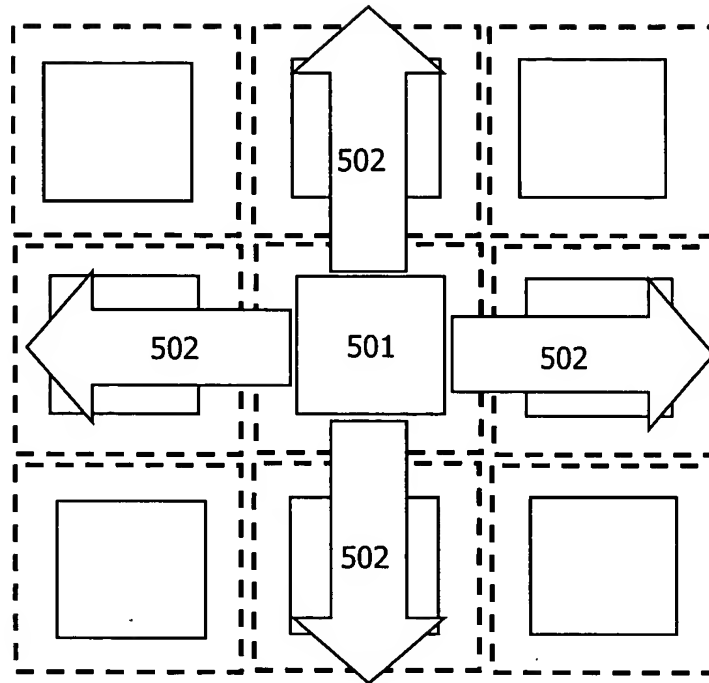


Figure 6

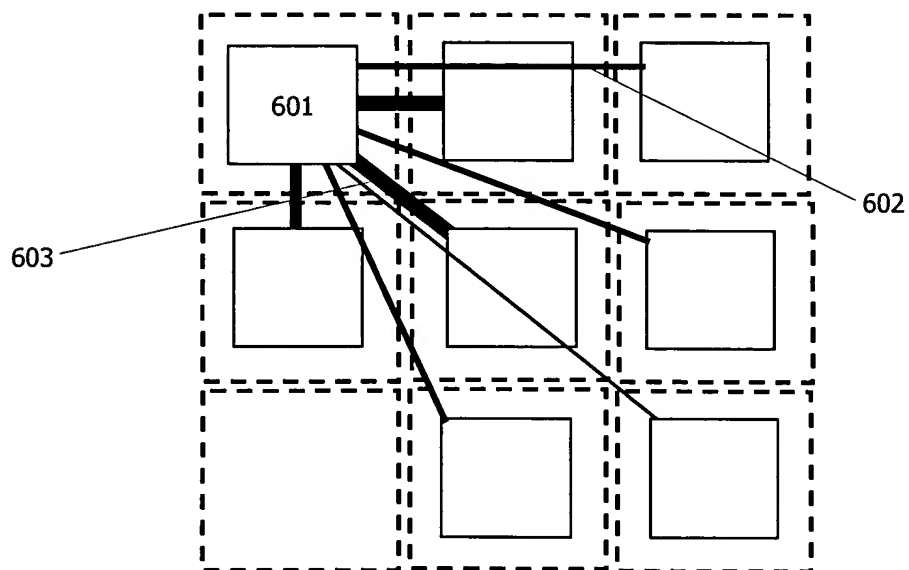


Figure 7

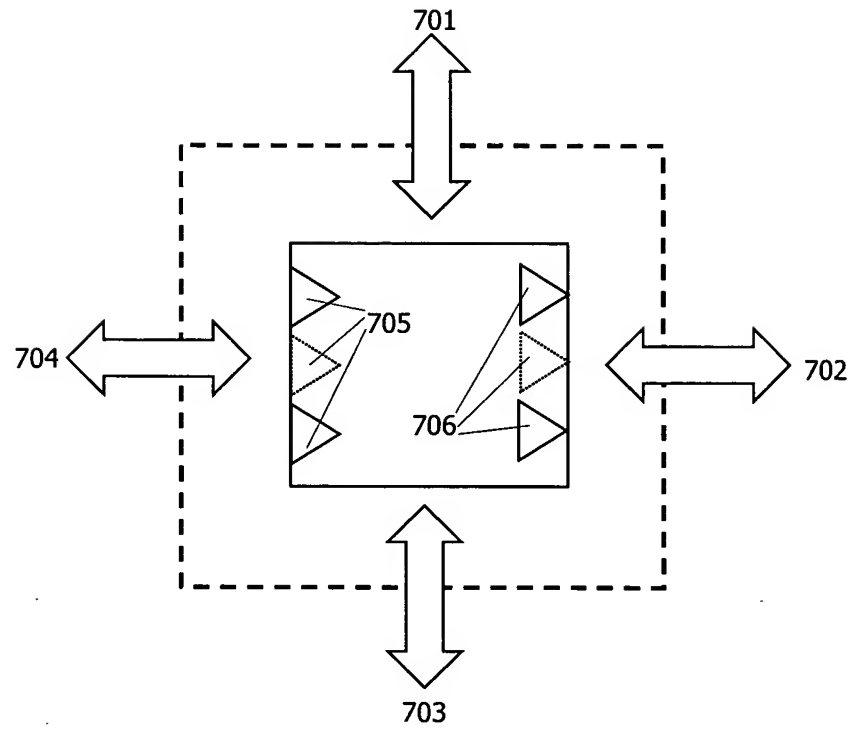


Figure 8

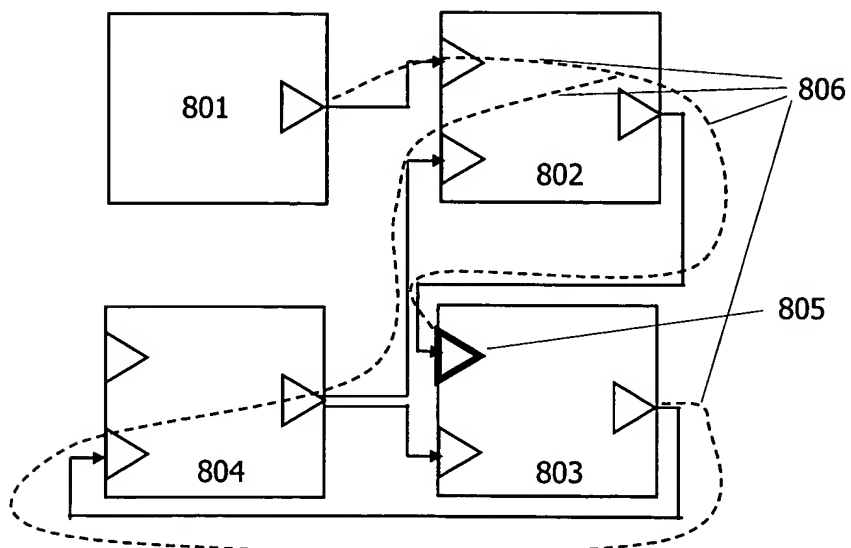


Figure 9

```

int sample[1040];      // input samples
int output[1024];      // filtered output samples
int coeff[16];         // filter coefficients
int i, *j, *k, sum;    // temporary variables
...
for (i = 0; i < 1024; i++) {
    sum = 0;
    for (j = &sample[i], k = &coeff[0]; k < &coeff[16]; j++; k++) {
        sum += (*j) * (*k);
    }
    output[i] = sum >> 16; // shift to take account of fixed point
}

```

Figure 10

INNER_LOOP:

```

LDR    r7, [r1], #4 ----- 1001
LDR    r8, [r0], #4 ----- 1002
CMP    r0, r14 ----- 1003
MLA    r2, r8, r7, r2 ----- 1004
BCC    INNER_LOOP ----- 1005

```

Figure 11

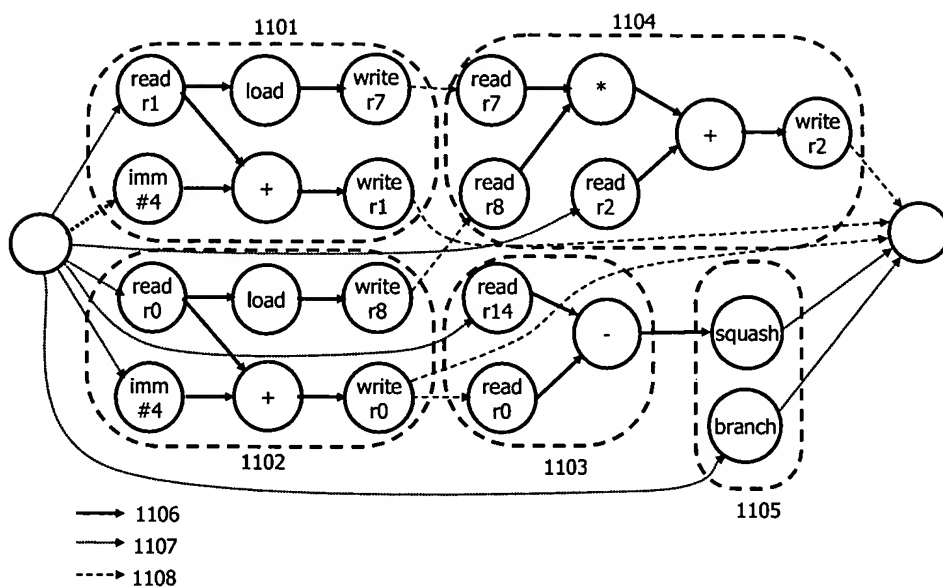


Figure 12

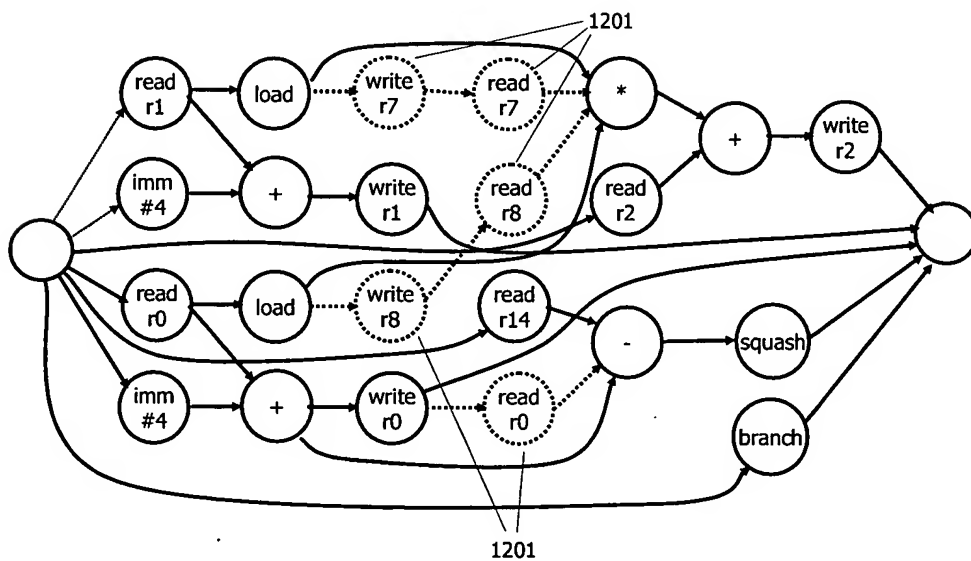


Figure 13

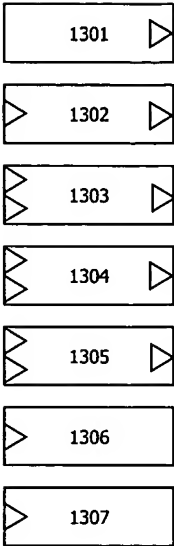


Figure 14

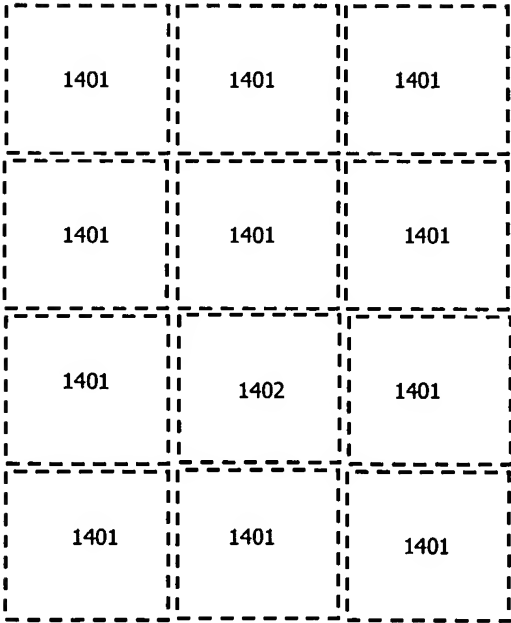


Figure 15

Results Operands	Arithmetic	Immediate	Memory	Multiply	Reg File
Arithmetic (Op1)	1	0	1	1	2
Arithmetic (Op2)	0	0	1	0	2
Branch	0	0	0	0	0
Memory	0	0	0	0	2
Multiply (Op1)	0	1	0	0	0
Multiply (Op2)	0	1	0	0	0
Reg File	3	0	0	0	0
Squash	1	0	0	0	0

Figure 16

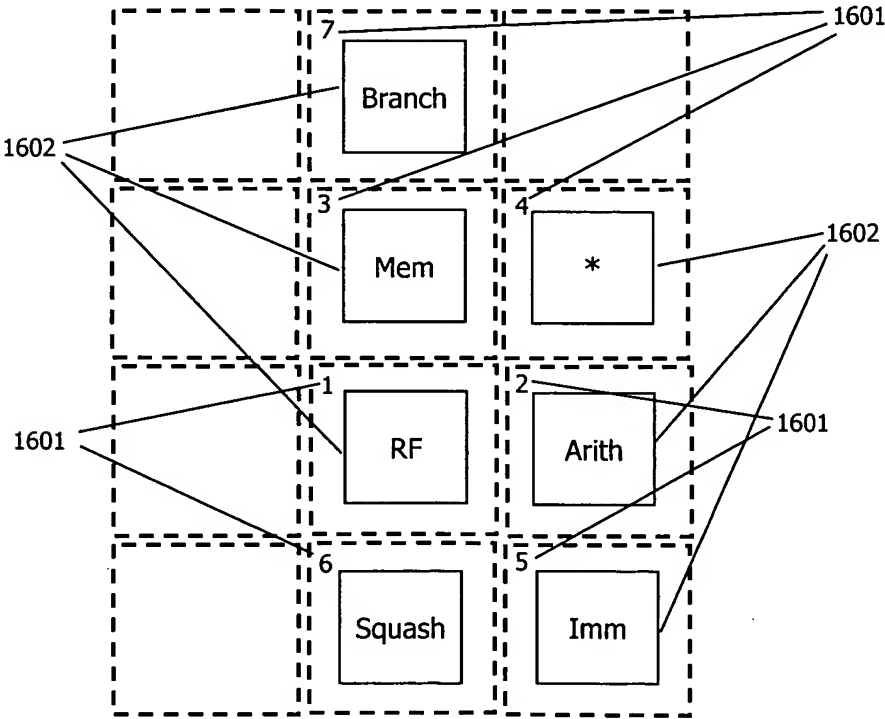


Figure 17

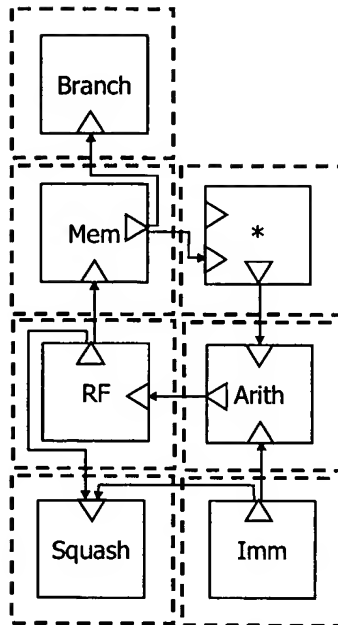


Figure 18

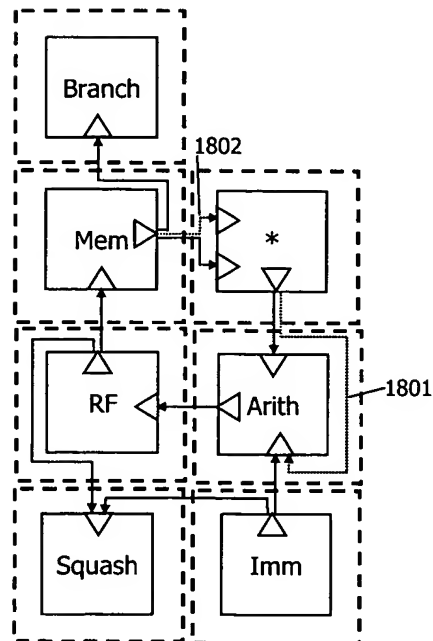


Figure 19

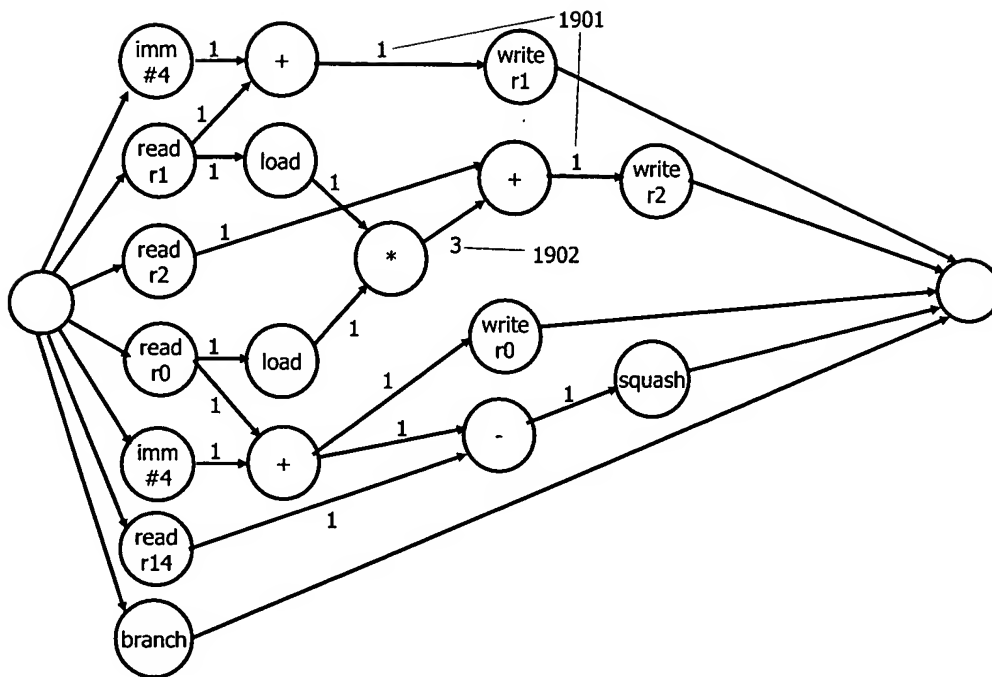


Figure 20

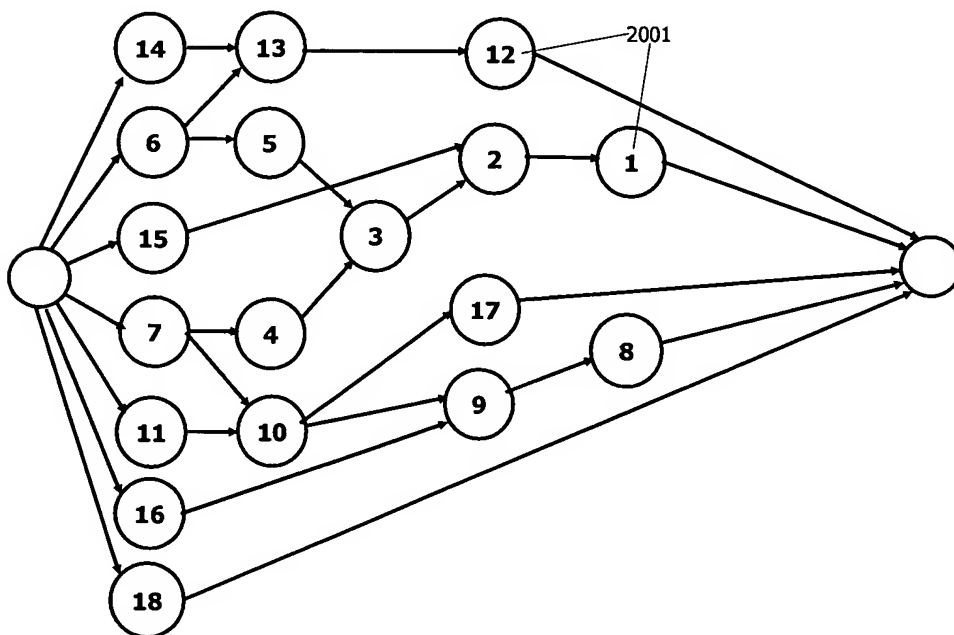


Figure 21

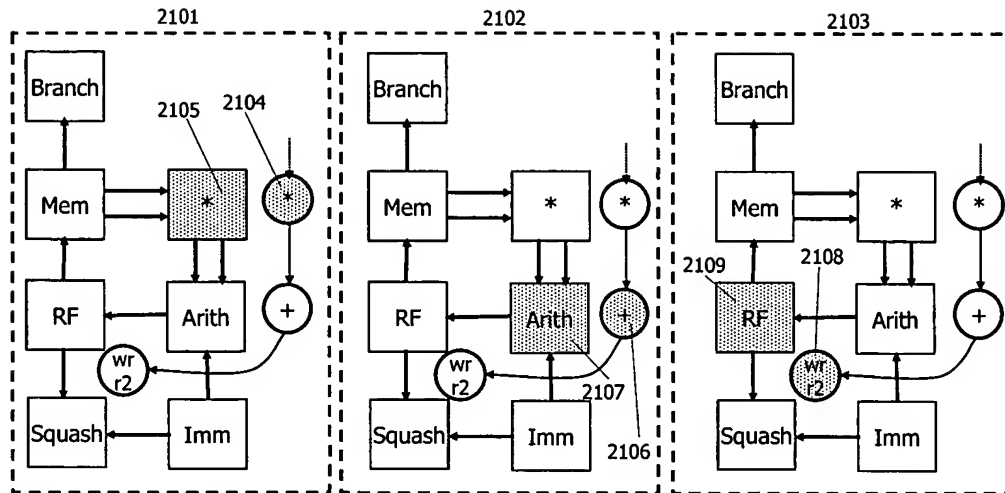


Figure 22

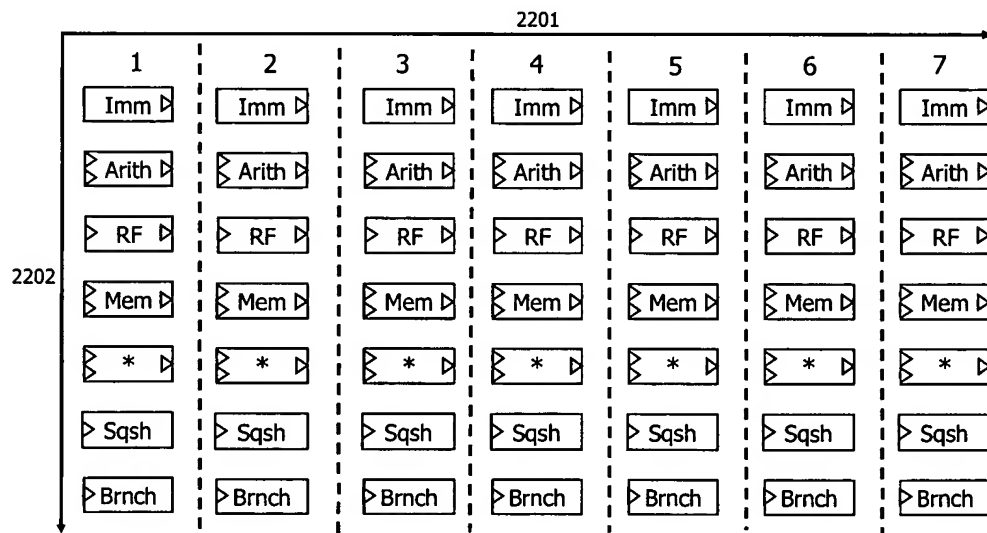


Figure 23

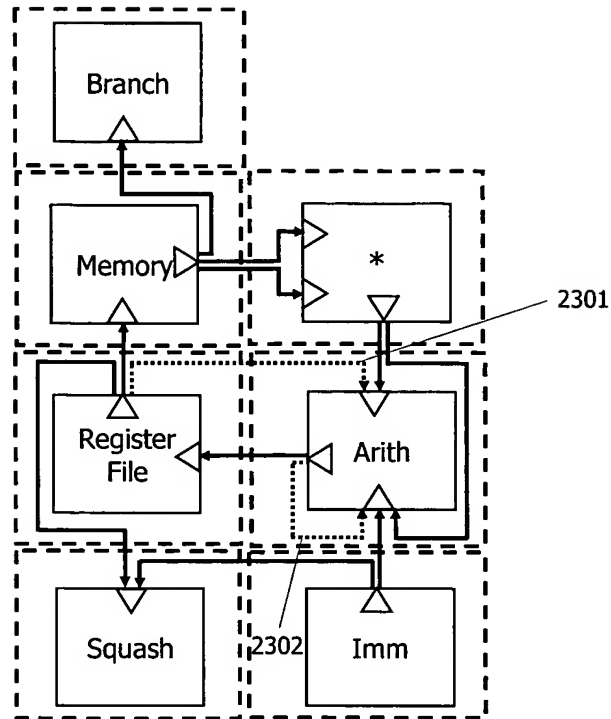


Figure 24

imm #4	imm #4	load mem	*	read r14	read r2	arith +	write r2
read r1	read r0	write r1	write r0		arith -		squash
branch	load mem	arith +					
	arith +						

